WebRTC 응용프로그램 구현

이 름 전상우 지도교수 노병희 멘 토 산고양

연구배경

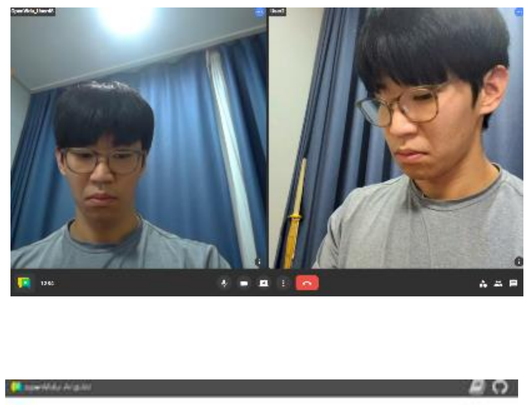
평소에 컴퓨터 통신에 관심이 있어서, 해당 프로젝트를 진행하기로 했다. Janus를 활용하여 WebRTC에서 미디어스트림의 흐름을 이해하고, 이를 활용하여 이를 활용한 응용프로그램을 제작하는 프로젝트이다.
오픈소스로 공개된 Janus를 활용해 다자간의 화상회의를 구현하는 계획을 수립하고, 코드를 수정해, client to server구조의 통신이 아닌 client to client 즉, 유사 P2P통신을 구축하는 것이 최종 목표이다.

연구진행과정

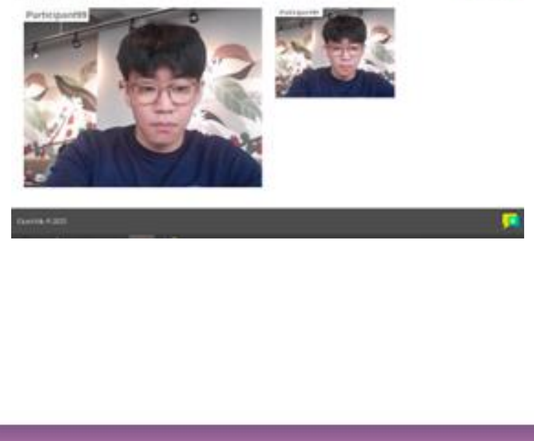
처음 프로젝트를 진행하는 과정에서 Janus는 os호환성, 코드 난잡화, 신버전의 모듈과 turn 서버의 호환성 등의 문제 때문에, 5주차에 OpenVidu를 활용하는 방식으로 프로젝트의 방향을 바꾸기도 하였다.
웹서버를 처음 활용해보기도 하고, 그 동안 과제에서나 몇 번씩 써보던 리눅스 환경에서 프로젝트를 진행하는 과정에서 익숙지 않은 환경이다 보니 처음 몇 주차 동안 환경에 적응하는 데에 시간을 많이 할애해서 다소 아쉬움이 남는 부분이다.

OpenVidu를 사용하였다. OpenVidu는 Janus와 다르게 Docker를 사용해, 가상환경에서 turn서버를 실행하기에, Janus보다 제약이 적었다. Janus는 javascript로 작성되어 있다. 다만 OpenVidu는 자바스크립트에서 파생된 언어 타입스크립트를 활용하여, 언어를 처음부터 숙달하는 난관이 있었다.
다행히 OpenVidu에서 제공하는 모듈의 튜토리얼 들을 따라하고 실행하면서, 어느정도 익숙해지는 데 도움이 됐다. 이후 튜토리얼 에서 사용했던 코드들을 분석해, 내 컴퓨터에서 입력되는 카메라와 오디오 파일을 분석하는 과정에 들어갔다. 코드가 상당히 길어, 분석하는 데에 시간을 많이 사용했다.

On premise version을 사용해 서로 다른 IP환경에서 현재 pc에서 웹서버를 가동해 화상회의를 시도해 보았고 정상적으로 작동하는 것을 확인하였다.



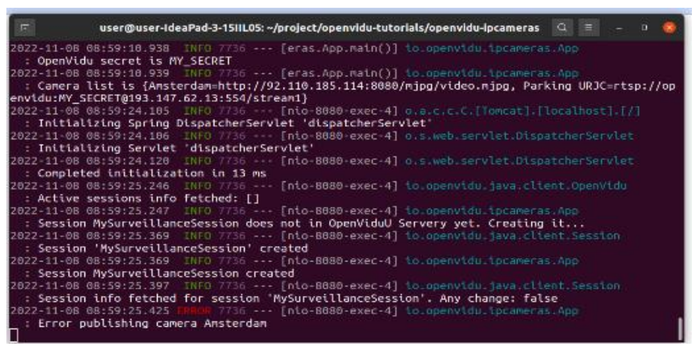
On premise version의 경우, 사전에 음성 녹화, 채팅, 화면공유, 음소거 등 줌에서 사용가능한 기능들을 제공한다. 다만 소스코드는 공개되지 않는다



개발자용 버전에서는 소스코드가 공개되어 있어, 개발자가 자유롭게 수정하고 조작할 수 있다. 다만 아주 간단한 기능만을 포함하고 있다.

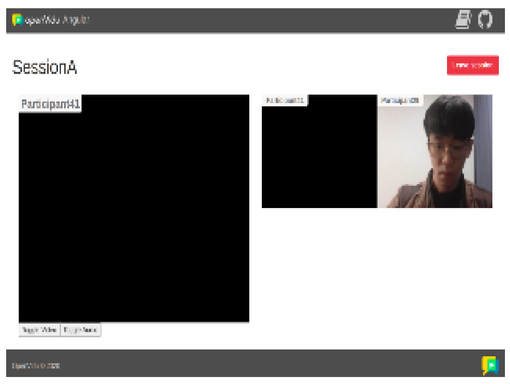
결과 및 분석

외부 IP를 통해 rtsp에서 영상을 받아와 스트리밍을 시도해 보았다. 서버는 해당 rtsp로 부터 영상을 받아와 클라이언트의 화면에 전송한다.



다만 아쉽게도 서버로그를 확인했을 시, 수신을 확인을 했지만, 출력이 이루어지지 않았다. 로그 상에서 해당 영상파일의 정보가 명확한 시점에서는 작성한 코드 단위에서 발생한 문제라 추측한다.

다만 클라이언트 단위에서 서버를 통해, 영상, 음성 송출을 차단하는데 성공했다.



개발자용 툴의 openvidu-angular 버전을 활용해 코드를 수정해 음소거, 비디오 비활성화를 구현했다. Angular는 구글에서 개발한 타입스크립트 프레임워크로 이번에 처음 활용해 본다는 점이 문제였다. Openvidu에서 제공하는 문서들은 대부분 배포판인 .ts 버전으로 작성돼 이를 이해하는데 어려움이 많았다. 해당 프로그램을 응용해, 특정 클라이언트가 전송하는 영상을 서버가 아닌 한 명의 클라이언트만 볼 수 있게 하는 기능을 추가할 예정이다.

아직 프로젝트가 끝나지 않은 현 시점에서 서버를 거치지 않고 클라이언트 간의 상호 통신을 이루기 위해서 연구가 더 필요한 시점이다.